

CLAIMS

What we claim is:

1 1. A system for updating electronic files comprising:
2 a first device for generating difference files that include coded differences
3 between an original version and a new version of an electronic file, wherein the first
4 device includes at least one component that reduces a size of the difference file by:
5 identifying differences between the original and the new versions
6 corresponding to address shifts resulting from at least one of code line deletion,
7 code line addition, and code line modification;
8 removing the identified differences in text sections common to the original
9 and the new versions by modifying target addresses of instructions of the original
10 version using at least one relationship between addresses of text sections of the
11 original version and corresponding text sections of the new version;
12 generating a modified version of the original file that includes the
13 instructions having modified target addresses; and
14 a second device that receives the difference file and generates a version of the
15 new file in the second device using the difference file.

1 2. The system of claim 1, wherein removing further includes:
2 identifying first and second text sections that are common between the original
3 and the new versions, wherein the first text section in the original version includes a first
4 calculable instruction and the second text section in the new version includes a second
5 calculable instruction;
6 identifying third and fourth text sections that are common between the original
7 and the new versions, wherein the third text section in the original version includes a first
8 target address that corresponds to the first calculable instruction, wherein the fourth text
9 section in the new version includes a second target address that corresponds to the second
10 calculable instruction;
11 generating a second instruction value from the second calculable instruction by
12 modifying the first instruction value using a first difference between starting addresses of

13 the third and fourth function units and using a second difference between starting
14 addresses of the first and second function units; and
15 replacing a first instruction value of the first calculable instruction with the second
16 instruction value.

1 3. The system of claim 1, further comprising removing the identified differences in
2 instructions including data pointers that point to data sections common to the original and
3 the new versions by modifying data pointer values in the original version, wherein
4 modifying data pointer values uses a difference between starting addresses of data
5 sections in the original version pointed to by a current data pointer and starting addresses
6 of corresponding data sections in the new version.

1 4. The system of claim 3, wherein removing identified differences in instructions
2 including data pointers further comprises:
3 identifying first and second code segments that are common between the original
4 and the new versions, wherein the first code segment in the original version includes a
5 first data pointer and the second code segment in the new version includes a second data
6 pointer;
7 identifying first and second data units that are common between the original and
8 the new versions, wherein the first data unit in the original version includes a first target
9 address that corresponds to the first data pointer, wherein the second data unit in the new
10 version includes a second target address that corresponds to the second data pointer;
11 generating a second data pointer value from the second data pointer by modifying
12 a first data pointer value using a difference between starting addresses of the first and
13 second data units; and
14 replacing a first data pointer value of the first data pointer with the second data
15 pointer value.

1 5. The system of claim 1, wherein the second device is at least one processor-based
2 device selected from among personal computers, portable computing devices, cellular
3 telephones, portable communication devices, and personal digital assistants.

1 6. The system of claim 1, wherein the first device transfers the difference file to the
2 second device using at least one coupling, wherein the at least one coupling is at least one
3 of a wireless coupling, a wired coupling, and a hybrid wireless/wired coupling.

1 7. An apparatus for use in generating difference files, comprising:
2 means for receiving an original version and a new version of an electronic file;
3 means for identifying units of code that are common to the original and the new
4 versions;
5 means for identifying instructions that are common to the units of code, wherein
6 the instructions include instruction values that direct processing to another portion of the
7 corresponding file;
8 means for generating a first instruction value from a first instruction of the
9 original version;
10 means for generating a second instruction value from a second instruction of the
11 new version, wherein the second instruction corresponds to the first instruction;
12 means for replacing the first instruction value of the first instruction with the
13 second instruction value; and
14 means for generating a modified version of the original file comprising the first
15 instruction with the second instruction value.

1 8. A method for reducing a size of a difference file that includes coded differences
2 between an original version and a new version of an electronic file, comprising:
3 identifying differences between the original and the new versions corresponding
4 to address shifts resulting from at least one of code line deletion, code line addition, and
5 code line modification;
6 removing the identified differences in text sections common to the original and
7 the new versions by modifying target addresses of instructions of the original version
8 using at least one relationship between addresses of text sections of the original version
9 and corresponding text sections of the new version; and

10 generating a modified version of the original file that includes the instructions
11 having modified target addresses.

1 9. The method of claim 8, wherein removing further includes:
2 identifying first and second text sections that are common between the original
3 and the new versions, wherein the first text section in the original version includes a first
4 calculable instruction and the second text section in the new version includes a second
5 calculable instruction;
6 identifying third and fourth text sections that are common between the original
7 and the new versions, wherein the third text section in the original version includes a first
8 target address that corresponds to the first calculable instruction, wherein the fourth text
9 section in the new version includes a second target address that corresponds to the second
10 calculable instruction;
11 generating a second instruction value from the second calculable instruction by
12 modifying the first instruction value using a first difference between starting addresses of
13 the third and fourth function units and using a second difference between starting
14 addresses of the first and second function units; and
15 replacing a first instruction value of the first calculable instruction with the second
16 instruction value.

1 10. The method of claim 8, further comprising removing the identified differences in
2 instructions including data pointers that point to data sections common to the original and
3 the new versions by modifying data pointer values in the original version, wherein
4 modifying data pointer values uses a difference between starting addresses of data
5 sections in the original version pointed to by a current data pointer and starting addresses
6 of corresponding data sections in the new version.

1 11. The method of claim 10, wherein removing identified differences in instructions
2 including data pointers further comprises:
3 identifying first and second code segments that are common between the original
4 and the new versions, wherein the first code segment in the original version includes a

5 first data pointer and the second code segment in the new version includes a second data
6 pointer;

7 identifying first and second data units that are common between the original and
8 the new versions, wherein the first data unit in the original version includes a first target
9 address that corresponds to the first data pointer, wherein the second data unit in the new
10 version includes a second target address that corresponds to the second data pointer;

11 generating a second data pointer value from the second data pointer; and

12 replacing a first data pointer value of the first data pointer with the second data
13 pointer value.

1 12. The method of claim 11, wherein generating the second data pointer value
2 includes modifying a first data pointer value using a difference between starting
3 addresses of the first and second data units.

1 13. The method of claim 10, further comprising generating a modified version of the
2 original file that includes the instructions having modified data pointers.

1 14. The method of claim 10, further comprising merging common data units to form
2 common data blocks.

1 15. The method of claim 14, wherein merging comprises:

2 combining first and second data units of the original version and combining
3 corresponding first and second data units of the new version to form a common data
4 block when the first data unit of the original version has a size equal to the first data unit
5 of the new version, an ending address of the first data unit of the original version is equal
6 to a starting address of the second data unit of the original version, and an ending address
7 of the first data unit of the new version is equal to a starting address of the second data
8 unit of the new version; and

9 repeating the combining operation for additional mergeable data units.

1 16. The method of claim 15, further comprising merging common text sections to
2 form common function blocks.

1 17. The method of claim 16, wherein merging comprises:
2 combining first and second text sections of the original version and combining
3 corresponding first and second text sections of the new version to form a common
4 function block when the first text section of the original version has a size equal to the
5 first text section of the new version, an ending address of the first text section of the
6 original version is equal to a starting address of the second text section of the original
7 version, and an ending address of the first text section of the new version is equal to a
8 starting address of the second text section of the new version; and
9 repeating the combining operation for additional mergeable text sections.

1 18. A method for performing file differencing, comprising:
2 receiving an original version and a new version of an electronic file;
3 identifying units of code that are common to the original and the new versions;
4 identifying instructions that are common to the units of code, wherein the
5 instructions include instruction values that relate to another portion of the corresponding
6 file;
7 decoding a first instruction value from a first instruction of the original version;
8 generating a second instruction value from a second instruction of the new
9 version, wherein the second instruction corresponds to the first instruction;
10 replacing the first instruction value of the first instruction with the second
11 instruction value; and
12 generating a modified version of the original file comprising the first instruction
13 with the second instruction value.

1 19. The method of claim 18, further comprising extracting the common units of code
2 from associated map files, wherein the common units of code include common function
3 units and common data units.

1 20. The method of claim 18, further comprising merging common function units of
2 the units of code to form common function blocks.

1 21. The method of claim 20, wherein merging comprises:
2 combining first and second function units of the original version and combining
3 corresponding first and second function units of the new version to form a common
4 function block when the first function unit of the original version has a size equal to the
5 first function unit of the new version, an ending address of the first function unit of the
6 original version is equal to a starting address of the second function unit of the original
7 version, and an ending address of the first function unit of the new version is equal to a
8 starting address of the second function unit of the new version; and
9 repeating the combining operation for additional mergeable function units.

1 22. The method of claim 20, further comprising encoding the common function
2 blocks.

1 23. The method of claim 18, further comprising merging common data units of the
2 units of code to form common data blocks.

1 24. The method of claim 23, wherein merging comprises:
2 combining first and second data units of the original version and combining
3 corresponding first and second data units of the new version to form a common data
4 block when the first data unit of the original version has a size equal to the first data unit
5 of the new version, an ending address of the first data unit of the original version is equal
6 to a starting address of the second data unit of the original version, and an ending address
7 of the first data unit of the new version is equal to a starting address of the second data
8 unit of the new version; and
9 repeating the combining operation for additional mergeable data units.

1 25. The method of claim 23, further comprising encoding the common data blocks.

1 26. The method of claim 18, wherein the units of code that are common to the
2 original and the new versions are common function units, wherein the first instruction
3 includes a first calculable instruction and the second instruction includes a second
4 calculable instruction, wherein generating the second instruction value includes
5 modifying the first instruction value using a first difference between starting addresses of
6 the common function units of the original and the new versions that include a target
7 address and using a second difference between the starting addresses of the common
8 function units that include the first and second calculable instructions.

1 27. The method of claim 18, wherein the units of code include function units. .

1 28. The method of claim 27, wherein generating the first instruction value includes
2 generating a difference between a current instruction address of the first calculable
3 instruction and a target address of the first calculable instruction.

1 29. The method of claim 27, wherein generating the second instruction value
2 includes:
3 generating a difference between a current instruction address of the first
4 calculable instruction and a target address of the first calculable instruction;
5 generating a difference between a start address of a first common function unit of
6 the new version and a start address of a corresponding first common function unit of the
7 original version; and
8 generating a difference between a start address of a second common function unit
9 of the new version and a start address of a corresponding second common function unit
10 of the original version.

1 30. The method of claim 18, wherein the units of code include data units and the
2 instructions include data pointers.

1 31. The method of claim 18, wherein the units of code that are common to the
2 original and the new versions are common data units, wherein the first instruction

3 includes a first data pointer and the second instruction includes a second data pointer,
4 wherein the first instruction value includes a first data pointer value and the second
5 instruction value includes a second data pointer value, wherein generating the second
6 instruction value includes generating a second data pointer value by modifying the first
7 data pointer value using a difference between starting addresses of the first and second
8 data units.

1 32. The method of claim 18, further comprising performing file differencing between
2 the modified version of the original file and the new file and generating a difference file.

1 33. The method of claim 32, further comprising:
2 transferring the difference file to a portable processing system; and
3 generating a version of the new file in the portable processing system using the
4 difference file.

1 34. The method of claim 33, further comprising:
2 merging common function units of the units of code to form common function
3 blocks;
4 merging common data units of the units of code to form common data blocks; and
5 encoding the common function blocks and the common data blocks for use in
6 generating the version of the new file in the portable processing system.

1 35. A method for determining differences between electronic files, comprising:
2 receiving an original version and a new version of an electronic file;
3 identifying first and second function units that are common between the original
4 and the new versions, wherein the first function unit in the original version includes a
5 first calculable instruction and the second function unit in the new version includes a
6 second calculable instruction;
7 identifying third and fourth function units that are common between the original
8 and the new versions, wherein the third function unit in the original version includes a
9 first target address that corresponds to the first calculable instruction, wherein the fourth

10 function unit in the new version includes a second target address that corresponds to the
11 second calculable instruction;
12 generating a second instruction value from the second calculable instruction;
13 replacing a first instruction value of the first calculable instruction with the second
14 instruction value; and
15 generating a modified version of the original file comprising the first instruction
16 with the second instruction value.

1 36. The method of claim 35, wherein generating the second instruction value includes
2 modifying the first instruction value using a first difference between starting addresses of
3 the third and fourth function units and using a second difference between starting
4 addresses of the first and second function units.

1 37. The method of claim 35, further comprising:
2 identifying first and second code segments that are common between the original
3 and the new versions, wherein the first code segment in the original version includes a
4 first data pointer and the second code segment in the new version includes a second data
5 pointer;
6 identifying first and second data units that are common between the original and
7 the new versions, wherein the first data unit in the original version includes a first target
8 address that corresponds to the first data pointer, wherein the second data unit in the new
9 version includes a second target address that corresponds to the second data pointer;
10 generating a second data pointer value from the second data pointer;
11 replacing a first data pointer value of the first data pointer with the second data
12 pointer value; and
13 generating a modified version of the original file comprising the first data pointer
14 with the second data pointer value.

1 38. The method of claim 37, wherein generating the second data pointer value
2 includes generating a difference between starting addresses of data sections in the

3 original version pointed to by a current data pointer and starting addresses of
4 corresponding data sections in the new version.

1 39. The method of claim 35, further comprising merging common function units to
2 form common function blocks.

1 40. The method of claim 35, further comprising merging common data units to form
2 common data blocks.

1 41. The method of claim 35, further comprising generating a difference file using the
2 modified version of the original file and the new file.

1 42. The method of claim 41, further comprising:
2 transferring the difference file to a portable processing system; and
3 generating a version of the new file in the portable processing system using the
4 difference file.

1 43. A computer readable medium including executable instructions which, when
2 executed in a processing system, reduce a size of a difference file that includes coded
3 differences between an original version and a new version of an electronic file by:
4 identifying differences between the original and the new versions corresponding
5 to address shifts resulting from at least one of code line deletion, code line addition, and
6 code line modification;
7 removing the identified differences in text sections common to the original and
8 the new versions by modifying target addresses of instructions of the original version
9 using at least one relationship between addresses of text sections of the original version
10 and corresponding text sections of the new version; and
11 generating a modified version of the original file that includes the instructions
12 having modified target addresses.